MANUAL DE UTILIZAÇÃO DO PLUGIN GERADOR DE PLUGINS DO TERRAVIEW

Versão TerraLib/TerraView: 3.6.0

Introdução

O plug-in de criação de plug-ins do TerraView foi desenvolvido para auxiliar os programadores TerraView/TerraLib na construção de plug-ins do TerraView. A construção de plug-ins, contém alguns passos comuns a todos os plug-ins desenvolvidos. Essa parte "burocrática" consiste na escrita do projeto do plug-in e de alguns arquivos que precisam ser escritos para que o plug-in funcione. Esses passos foram automatizados pelo plug-in de criação de plug-ins de forma que o desenvolvedor se preocupe apenas com a implementação das funcionalidades.

Esse manual detalha a criação de plug-ins usando o plug-in de criação de plug-ins. Nas próximas sessões serão apresentados os arquivos gerados por este plug-in, o manual de uso do plug-in de plug-ins e um exemplo de construção de um plug-in usando o plug-in de criação.

Os arquivos gerados

Para o desenvolvimento de plug-ins, primeiramente, são necessários, no mínimo, cinco arquivos: o arquivo de projeto, arquivos de cabeçalho e código de inicialização do plug-in (2), arquivos de cabeçalho e código da classe de plug-in (2). A seguir serão explicados cada arquivo listado e sua funcionalidade.

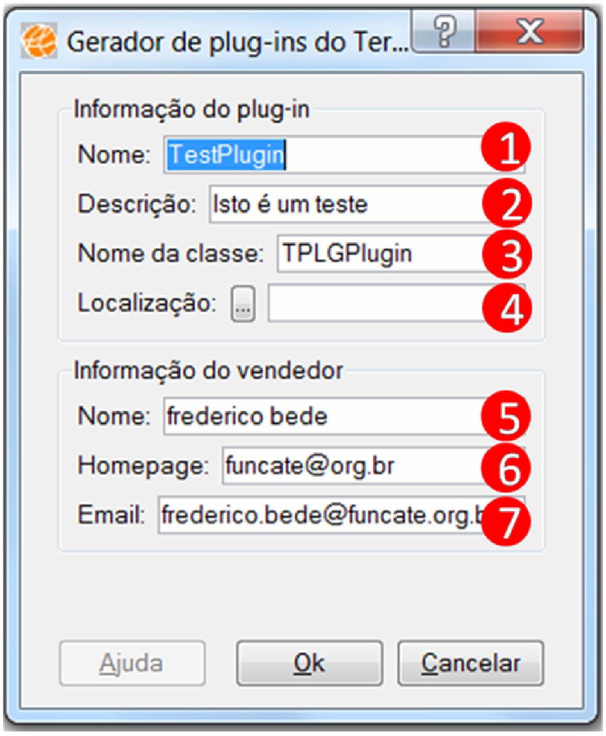
- Projeto do plug-in: contém as definições de um projeto de biblioteca de software do Qt(.pro). Esse arquivo contém definições dos projetos, como arquivos e flags que serão usados na compilação do projeto C++.
- Cabeçalho do arquivo de inicialização do plug-in: contém definições de duas funções que serão usadas pelo plug-in – inicialização (na carga do plug-in) e finalização (na destruição do plug-in). Além disso contém os parâmetros do plug-in, como seu nome, desenvolvedor, versão, homepage e etc.
- Código do arquivo de inicialização do plug-in: implementação das funções de inicialização e finalização do plug-in. A implementação dessas funções é padrão para todos os plug-ins: a inicialização instancia um objeto de plug-in e a finalização o destrói.

- Cabeçalho do arquivo da classe de plug-in: contém a definição da classe de plug-in. Esta classe herda de uma outra contida no TerraView chamada TeAbstractPlugin, e da classe QObject, para que se possa registrar slots Qt e usá-los no plug-in. A classe de plug-in contém dois métodos que devem ser implementados: init() e end(). Essas funções serão chamadas durante a inicialização e destruição do plug-in respectivamente.

- Arquivo de código da classe plug-in: contém a implementação da classe do plug-in. Os métodos init() e end() deverão ser codificados para inicializar e finalizar o plug-in respectivamente.

O plug-in de criação de plug-ins do TerraView

O plug-in de criação de plug-ins é composto de uma interface simples que disponibiliza ao desenvolvedor uma interface para entrada de dados do plug-in a ser gerado. A figura a seguir apresenta a interface.



Os números apresentados à direita dentro de um círculo vermelho serão usados para identificação e explicação dos parâmetros que devem ser fornecidos pelo desenvolvedor. Os quatro primeiros parâmetros são relacionados ao plug-in e os outros ao desenvolvedor. Os parâmetros são os seguintes:

1. *Nome*: nome do plug-in. Esse nome será dado à pasta que contém o plug-in, ao projeto e será prefixo dos nomes dos arquivos de inicialização do plug-in.

2. *Descrição*: contém uma breve descrição do plug-in.

3. *Nome da classe*: nome da classe de plug-in. Esse nome será prefixo dos nomes dos arquivos de definição e implementação da classe de plug-in.

4. *Localização*: diretório onde será gerado o plug-in. Note que esse diretório deve ser corresponder ao diretório *terraViewPlugins* contido na pasta *terralib* onde o desenvolvedor baixou o código fonte da TerraLib/TerraView. Clique no botão e selecione o diretório.

5. *Nome*: nome do desenvolvedor. *Homepage*: homepage do desenvolvedor.

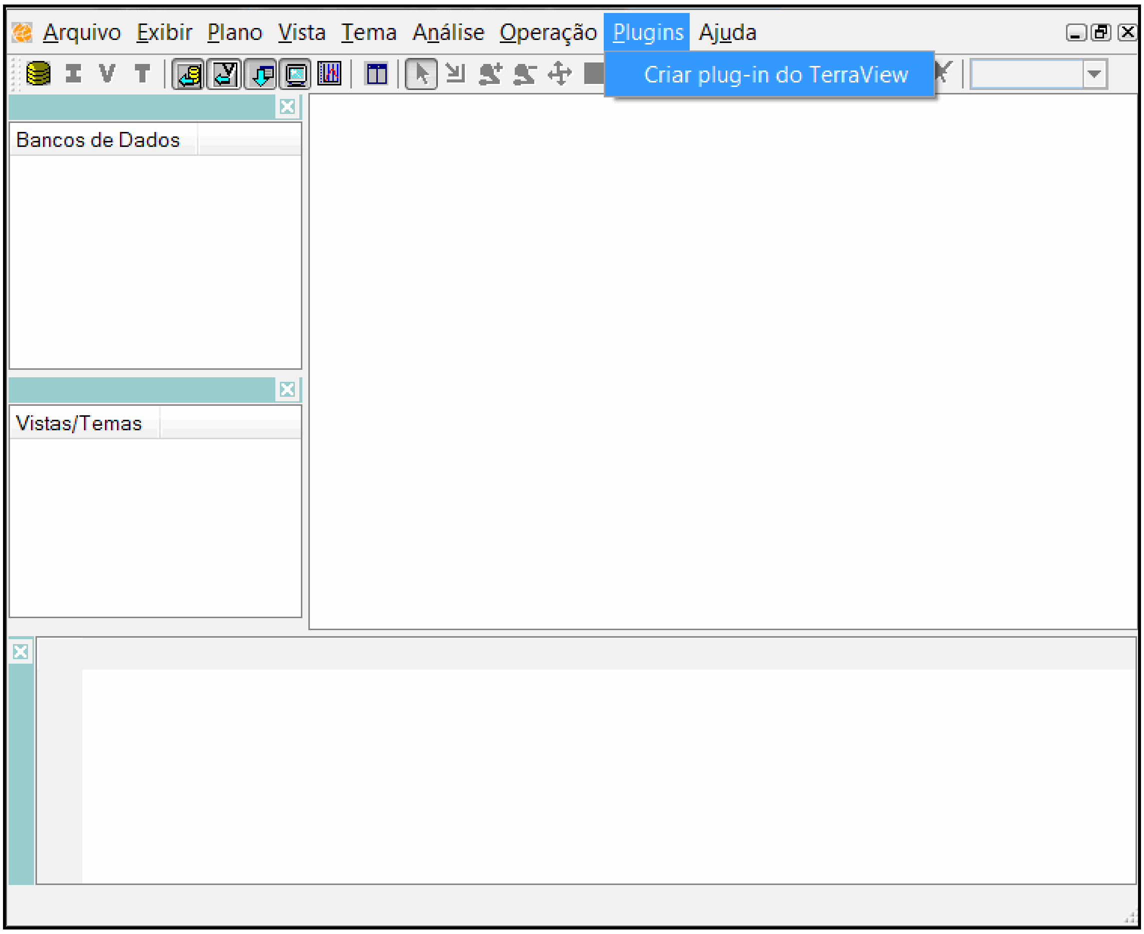
6. *Email*: email do desenvolvedor.

Todos os parâmetros devem ser fornecidos. Lembrando que a localização deve ser o diretório *terraViewPlugins* contido na pasta *terralib* baixada pelo desenvolvedor. Os parâmetros de informação do vendedor podem ser preenchidos com valores não válidos, ou seja, uma homepage que não existe ou email que não existe, etc. Esses parâmetros são apenas informativos e não causam impacto algum sobre o desenvolvimento do plug-in.

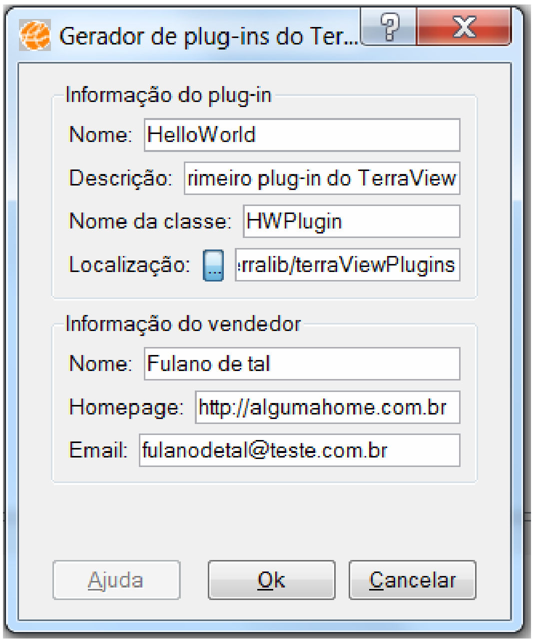
Criando um plug-in

Nesta sessão será criado um plug-in HelloWorld mostrando como usar o plug-in de criação do TerraView.

Primeiramente o desenvolvedor deve baixar a TerraLib do repositório do INPE, dentro de uma pasta qualquer no sistema. Instruções para acesso e obtenção da TerraLib podem ser encontradas na página principal da TerraLib em http://www.dpi.inpe.br/terralib. Depois de compilados terraView e plug-in de criação devemos executar o plug-in de criação. Para executá-lo vá em "Plugins" no menu principal do TerraView e selecione a opção de "Criar plug-in do TerraView" como mostrado na figura abaixo.

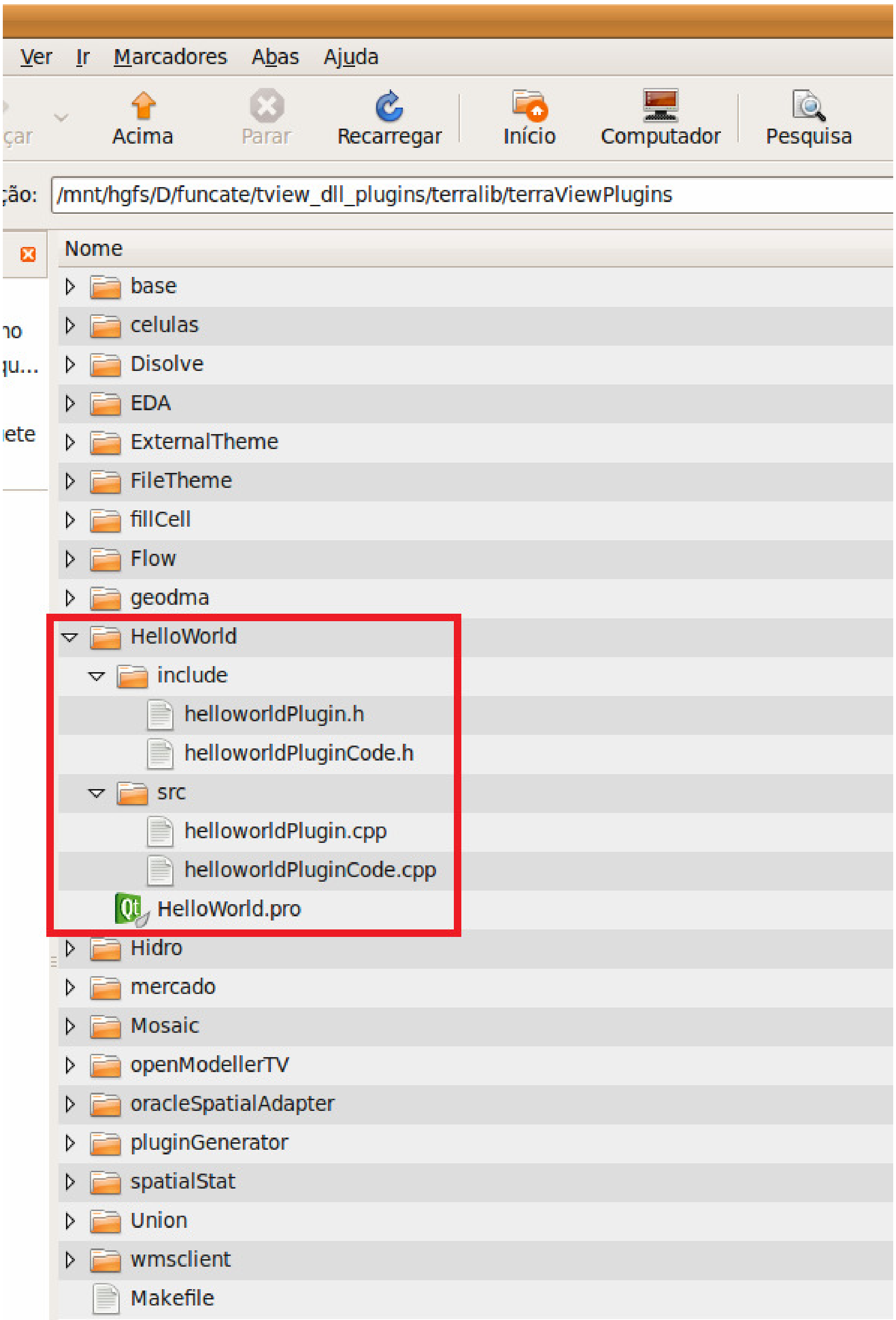


A tela de criação de plug-ins será exibida preencha os parâmetros de criação conforme mostrados na figura a seguir. Lembrando que o plug-in deve ser criado sob a pasta *terraViewPlugins* dentro da pasta *terralib* conforme já falado.



Com os parâmetros preenchidos, clique no botão "Ok". Uma mensagem de sucesso deverá ser exibida. Pronto! O esqueleto do plug-in já foi gerado. A princípio o projeto é para Linux, mas é possível e muito simples convertê-lo para Windows. As instruções para conversão em projeto Windows são escritas junto com o projeto do plug-in como comentários. Essas instruções serão explicadas nas próximas sessões deste tutorial.

Uma vez executado o plug-in de criação será criada a seguinte estrutura de diretórios:



Essa é a estrutura do diretório "terraViewPlugins" com o novo plug-in criado. Note que foram criados a pasta "HelloWorld", o projeto "HelloWorld.pro", as subpastas "include", com dois arquivos de cabeçalho, e "src", com dois arquivos de implementação. Os cinco arquivos mencionados no inicio do tutorial estão divididos nas subpastas. Isso já é suficiente para compilar os arquivos do plug-in. O próximo passo será inserir um item no menu de plug-ins do TerraView.

Inserindo funcionalidades no plug-in

Para esse exemplo construiremos um plug-in simples, que somente apresenta uma mensagem de boas vindas quando for executado. Para isso, abra o arquivo "helloworldPlugin.h", que existe na subpasta "include". Defina um novo slot Qt, na classe de plug-in, chamado execPlugin(). O arquivo se parecerá com o seguinte:

```
1  #ifndef __HELLOWORLD_PLUGIN_H_
2  #define __HELLOWORLD_PLUGIN_H_
3
4  #include <TViewAbstractPlugin.h>
5
6  //Qt include files
7  #include <qobject.h>
8
9  class HWPlugin: public QObject, public TViewAbstractPlugin
10 {
11     Q_OBJECT
12
13 public:
14     /** \brief Constructor.
15      *  \param params Plugin parameters.
16      */
17     HWPlugin(PluginParameters* params);
18
19     /** \brief Destructor.
20      */
21     ~HWPlugin();
22
23     /** \brief Initializer method.
24      */
25     void init();
26
27     /** \brief Finalizer method.
28      */
29     void end();
30
31 protected slots:
32
33     void execPlugin();
34 };
35
36 #endif //__HELLOWORLD_PLUGIN_H_
```

A linha 33 mostra a definição do novo slot Qt.

Agora precisamos implementar a classe de plug-in de forma a inserir a opção no menu do TerraView e conectá-lo ao novo slot criado. Para isso insira, no arquivo de implementação do plug-in, (que neste exemplo é helloworldPlugin.cpp), as linhas necessárias para que o arquivo se pareça com o seguinte:

```
1  #include <helloworldPlugin.h>
2  //Qt include files
3  #include <qmessagebox.h>
4
5  HWPlugin::HWPlugin(PluginParameters* params):
6  QObject(),
7  TViewAbstractPlugin(params)
8  {
9      loadTranslationFiles("helloworld_");
10 }
11
12 HWPlugin::~HWPlugin()
13 {
14     end();
15 }
16
17 void HWPlugin::init()
18 {
19     TerraViewBase* tview = getMainWindow();
20
21     if(tview)
22     {
23         QPopupMenu* mnu = getPluginsMenu();
24
25         if(mnu != 0)
26         {
27             mnu->insertItem(tr("Hello World plug-in"), this, SLOT(execPlugin()));
28         }
29     }
30 }
31
32 void HWPlugin::end()
33 {
34 }
35
36 void HWPlugin::execPlugin()
37 {
38     TerraViewBase* tview = getMainWindow();
39
40     if(tview)
41         QMessageBox::information(tview, tr("Hello world plug-in"),
42         tr("Hello world plug-in works fine!"));
43 }
```

O método init() será implementado de forma a inserir uma opção no menu Plugins do TerraView com o texto “Hello World plug-in”. Esse código pode ser verificado nas linhas 19 a 29 da imagem anterior. Esse código insere uma entrada no menu de plug-ins do TerraView que, quando clicada, chama o slot criado execPlugin(). A função execPlugin apenas apresenta uma mensagem na tela, e pode ser verificada nas linhas 38 a 42 da figura anterior. Feito isso é só compilar o plug-in e executar o TerraView para ver o novo plug-in funcionando. Note que existe no projeto (arquivo .pro) uma definição para

gerar o arquivo de tradução do qt. Esse arquivo é usado para internacionalização do software pelo qt. Esse arquivo será usado pelo plug-in se ele estiver disponível. Para disponibilizá-lo, copie o arquivo gerado pelo QLinguist (ver manual do Qt para informações sobre internacionalização) para dentro do diretório de plug-ins do TerraView, onde o plug-in(.so no caso do Linux, .dll no caso do windows) está localizado.

Criando o projeto Windows

As regras para geração dos projetos Windows são escritas como comentários no arquivo de projeto. A imagem a seguir mostra o comentário.

```
#----------------------------------------------------------------------------------------------------------
#If you want to create a visual studio project, use the commmand "qmake -t vclib -spec win32-msvc.net".
#For debug version add the expression "TE_PROJECT_TYPE=DEBUG" to previous command.
#----------------------------------------------------------------------------------------------------------
include( ../base/base.pro )
TARGET = HelloWorld
TEMPLATE     = lib
LANGUAGE     = C++
TRANSLATIONS = helloworld_pt.ts
UI_DIR = src/ui
MOC_DIR = src/moc
INCLUDEPATH += ./include
HEADERS +=  ./include/helloworldPlugin.h \
    ./include/helloworldPluginCode.h
SOURCES +=  ./src/helloworldPlugin.cpp \
    ./src/helloworldPluginCode.cpp
FORMS +=
```

O destaque é para um comentário que vem no arquivo de projeto do plug-in gerado. São instruções para criação do projeto Windows. Para realizá-lo, basta abrir uma janela prompt, entrar no diretório onde se localiza o arquivo de projeto do plug-in (para este exemplo o arquivo é o HelloWorld.pro) e digitar o comando "qmake –t vclib –spec win32-msvc.net". Esse comando cria um projeto do visual Studio com configurações de release. Se for necessária a versão debug, basta acrescentar ao final do comando anterior a expressão "TE_PROJECT_TYPE=DEBUG". Depois disso é só abrir o arquivo .vcproj gerado usando o visual Studio, compilar e rodar o plug-in.

Essa rotina de geração do projeto Windows pode ser disparada tanto do Windows quanto do Linux.